

Retention Stage: Churn Prediction Model

Churn prediction models are crucial in the retention phase to identify customers who are likely to leave the bank. By proactively addressing their needs and concerns, the bank can implement targeted retention strategies to reduce churn and maintain a loyal customer base.

Churn Prediction Model

Objective:

- Predict which customers are likely to churn (leave the bank) and take preemptive actions to retain them.
- Enhance customer loyalty and reduce churn rate by addressing issues before customers decide to leave.

Methodology:

1. Data Preparation:

- Collect and preprocess customer data, including demographics, transaction history, product usage, and engagement metrics.
- Data sources include transactional logs, customer service interactions, and historical churn data.

2. Attribute Selection:

- **Raw Attributes:**

- Customer ID
- Age
- Income
- Occupation
- Credit score (CIBIL score)
- Product holdings (e.g., savings account, loan, credit card)
- Transaction history (frequency, recency, monetary value)

- Customer interactions (e.g., website visits, customer service interactions)
- Complaint logs and resolution times
- Tenure with the bank
- **Derived Attributes:**
 - Engagement score: Measure of overall engagement with the bank's services.
 - Customer satisfaction score: Derived from surveys or feedback forms.
 - Churn score: Probability score indicating likelihood of churn.
 - Product utilization rate: Frequency and extent of product usage.
- **Target Variable:**
 - Churn status (binary: 1 if the customer churned, 0 otherwise)

3. Model Selection:

- **Classification Algorithms:**
 - Logistic Regression
 - Random Forest
 - Gradient Boosting Machines (GBM)
 - Support Vector Machine (SVM)
 - Neural Networks
- **Evaluation Metrics:**
 - Accuracy
 - Precision
 - Recall
 - F1-Score
 - Area Under the ROC Curve (AUC-ROC)

Implementation Steps

1. Data Collection and Preparation:

```

import pandas as pd
from sklearn.preprocessing import StandardScaler, OneHot
Encoder
from sklearn.model_selection import train_test_split

# Sample customer data
data = {
    'customer_id': [1, 2, 3, 4, 5],
    'age': [25, 45, 35, 50, 23],
    'income': [50000, 120000, 75000, 100000, 45000],
    'cibil_score': [700, 800, 750, 780, 690],
    'product_holdings': ['savings,loan', 'savings,credit
card', 'savings,loan', 'savings', 'loan'],
    'transaction_frequency': [15, 50, 25, 30, 10],
    'average_transaction_value': [2000, 5000, 3000, 400
0, 1500],
    'engagement_score': [3, 5, 4, 4, 2],
    'tenure': [2, 10, 5, 8, 1],
    'churn_status': [0, 1, 0, 0, 1]
}
df = pd.DataFrame(data)

# Encode categorical variables
encoder = OneHotEncoder()
encoded_products = encoder.fit_transform(df[['product_ho
ldings']]).toarray()
df = df.join(pd.DataFrame(encoded_products, columns=enco
der.get_feature_names_out(['product_holdings']))).drop
('product_holdings', axis=1)

# Standardize numerical features
scaler = StandardScaler()
numerical_features = ['age', 'income', 'cibil_score', 't
ransaction_frequency', 'average_transaction_value', 'eng
agement_score', 'tenure']
df[numerical_features] = scaler.fit_transform(df[numeric
al_features])

```

```
# Split data into train and test sets
X = df.drop(['customer_id', 'churn_status'], axis=1)
y = df['churn_status']
X_train, X_test, y_train, y_test = train_test_split(X,
y, test_size=0.2, random_state=42)
```

2. Model Training and Evaluation:

Random Forest Classifier:

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, roc_auc_score

# Train Random Forest model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Predict on test data
y_pred = model.predict(X_test)
y_prob = model.predict_proba(X_test)[:, 1]

# Evaluate model performance
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
auc_roc = roc_auc_score(y_test, y_prob)

print(f'Accuracy: {accuracy}')
print(f'Precision: {precision}')
print(f'Recall: {recall}')
print(f'F1-Score: {f1}')
print(f'ROC AUC: {auc_roc}')
```

Variables for Churn Prediction

1. Customer Attributes:

- **Customer ID:** Unique identifier for each customer.
- **Age:** Different age groups may have varying churn rates.
- **Income:** Financial stability may influence customer loyalty.
- **Occupation:** Reflects lifestyle and financial needs.
- **Credit Score:** Higher scores may correlate with lower churn rates.
- **Product Holdings:** Number and type of products owned by the customer.

2. Transactional and Behavioral Data:

- **Transaction Frequency:** Frequency of transactions can indicate engagement.
- **Average Transaction Value:** Monetary value of transactions.
- **Engagement Score:** Overall interaction and activity with the bank.
- **Tenure:** Duration of the customer's relationship with the bank.
- **Complaint Logs:** History of customer complaints and resolution times.

3. Derived Attributes:

- **Churn Score:** Probability of the customer leaving the bank.
- **Engagement Score:** Comprehensive measure of customer activity.
- **Customer Satisfaction Score:** Derived from feedback or survey responses.
- **Product Utilization Rate:** Frequency and extent of product usage.

4. Target Variable:

- **Churn Status:** Binary indicator of whether the customer has churned (1) or not (0).

Conclusion

By implementing a churn prediction model, the bank can proactively identify at-risk customers and implement targeted retention strategies. This approach helps in reducing churn rates, maintaining customer loyalty, and enhancing overall customer satisfaction. Using advanced machine learning techniques and a comprehensive set of attributes, the bank can effectively predict and address customer churn, ensuring a stable and loyal customer base.